```
NNN        NNN   IIIIIIIIII                   CCCCCCCCCCC   NNN        NNN   FFFFFFFFFFFFFF
NNN        NNN   IIIIIIIIII                   CCCCCCCCCCC   NNN        NNN   FFFFFFFFFFFFFF
NNN        NNN   IIIIIIIIII                   CCCCCCCCCCC   NNN        NNN   FFFFFFFFFFFFFF
NNN        NNN       III             CCC                   NNN        NNN   FFF
NNN        NNN       III            CCC                    NNN        NNN   FFF
NNNNNN     NNN       III            CCC                    NNNNNN     NNN   FFF
NNNNNN     NNN       III            CCC                    NNNNNN     NNN   FFF
NNNNNN     NNN       III            CCC                    NNNNNN     NNN   FFF
NNN   NNN  NNN       III            CCC                    NNN   NNN  NNN   FFFFFFFFFFF
NNN   NNN  NNN       III            CCC                    NNN   NNN  NNN   FFFFFFFFFFF
NNN    NNN NNN       III            CCC                    NNN    NNN NNN   FFFFFFFFFF
NNN     NNNNNN       III            CCC                    NNN     NNNNNN   FFF
NNN     NNNNNN       III            CCC                    NNN     NNNNNN   FFF
NNN     NNNNNN       III            CCC                    NNN     NNNNNN   FFF
NNN        NNN       III            CCC                    NNN        NNN   FFF
NNN        NNN       III            CCC                    NNN        NNN   FFF
NNN        NNN       III             CCC                   NNN        NNN   FFF
NNN        NNN   IIIIIIIIII            CCCCCCCCCCC         NNN        NNN   FFF
NNN        NNN   IIIIIIIIII            CCCCCCCCCCC         NNN        NNN   FFF
NNN        NNN   IIIIIIIIII            CCCCCCCCCCC         NNN        NNN   FFF
```

```
CCCCCCC  NN      NN  FFFFFFFFFF  SSSSSSSS  HH      HH    000000    WW            WW
CCCCCCCC  NN      NN  FFFFFFFFFF  SSSSSSS   HH      HH    000000    WW            WW
CC        NN      NN  FF          SS        HH      HH  00      00  WW            WW
CC        NN      NN  FF          SS        HH      HH  00      00  WW            WW
CC        NNNN    NN  FF          SS        HH      HH  00      00  WW            WW
CC        NNNN    NN  FF          SS        HH      HH  00      00  WW            WW
CC        NN  NN  NN  FFFFFFFF    SSSSSS    HHHHHHHHHH  00      00  WW            WW
CC        NN  NN  NN  FFFFFFFF    SSSSSS    HHHHHHHHHH  00      00  WW            WW
CC        NN    NNNN  FF              SS    HH      HH  00      00  WW    WW      WW
CC        NN    NNNN  FF              SS    HH      HH  00      00  WW    WW      WW
CC        NN      NN  FF              SS    HH      HH  00      00  WWWW    WWWW
CC        NN      NN  FF              SS    HH      HH  00      00  WWWW    WWWW      ....
CCCCCCC   NN      NN  FF          SSSSSSSS  HH      HH    000000    WW      WW        ....
CCCCCCC   NN      NN  FF          SSSSSSSS  HH      HH    000000    WW      WW        ....
```

```
LL             IIIIII       SSSSSSSS
LL             IIIIII       SSSSSSSS
LL               II       SS
LL               II       SS
LL               II       SS
LL               II       SS
LL               II         SSSSSS
LL               II         SSSSSS
LL               II             SS
LL               II             SS
LL               II             SS
LL               II             SS
LLLLLLLLLL     IIIIII       SSSSSSSS
LLLLLLLLLL     IIIIII       SSSSSSSS
```

```
   1   0001  0 %TITLE  'DECnet Ethernet Configurator Module'
   2   0002  0 MODULE CNFSHOW               (
   3   0003  0                              LANGUAGE (BLISS32),
   4   0004  0                              IDENT = 'V04-000'
   5   0005  0                              ) =
   6   0006  1 BEGIN
   7   0007  1
   8   0008  1 !
   9   0009  1 !****************************************************************************
  10   0010  1 !*                                                                          *
  11   0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
  12   0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
  13   0013  1 !*  ALL RIGHTS RESERVED.                                                    *
  14   0014  1 !*                                                                          *
  15   0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
  16   0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH LICENSE  AND WITH THE    *
  17   0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
  18   0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
  19   0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
  20   0020  1 !*  TRANSFERRED.                                                            *
  21   0021  1 !*                                                                          *
  22   0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
  23   0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
  24   0024  1 !*  CORPORATION.                                                            *
  25   0025  1 !*                                                                          *
  26   0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
  27   0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
  28   0028  1 !*                                                                          *
  29   0029  1 !*                                                                          *
  30   0030  1 !****************************************************************************
  31   0031  1 !
  32   0032  1
  33   0033  1 !++
  34   0034  1 ! FACILITY:     DECnet Configurator Module (NICONFIG)
  35   0035  1 !
  36   0036  1 ! ABSTRACT:
  37   0037  1 !
  38   0038  1 !      This module contains the routines to return information on a
  39   0039  1 !      SHOW request generated by an NCP> SHOW MODULE CONFIGUTOR command.
  40   0040  1 !
  41   0041  1 ! ENVIRONMENT: VAX/VMS Operating System
  42   0042  1 !
  43   0043  1 ! AUTHOR:       Bob Grosso,     CREATION DATE:  13-Oct-1982
  44   0044  1 !
  45   0045  1 ! MODIFIED BY:
  46   0046  1 !
  47   0047  1 !--
```

CNFSHOW
V04-000

DECnet Ethernet Configurator Module
Definitions

H 2
16-Sep-1984 02:05:37   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:49:54   [NICNF.SRC]CNFSHOW.B32;1

Page 2
     (2)

```
  49    0048  1  %SBTTL 'Definitions'
  50    0049  1
  51    0050  1  !
  52    0051  1  !  INCLUDE FILES:
  53    0052  1  !
  54    0053  1
  55    0054  1  LIBRARY 'SYS$LIBRARY:STARLET';   ! VMS common definitions
  56    0055  1
  57    0056  1  LIBRARY 'SHRLIB$:NMALIBRY';      ! NICE code definitions
  58    0057  1
  59    0058  1  REQUIRE 'LIB$:CNFDEF.R32';
  60    0149  1
  61    0150  1  REQUIRE 'SRC$:CNFPREFIX.REQ';
  62    0247  1
  63    0248  1
  64    0249  1  !
  65    0250  1  !  BUILTIN functions
  66    0251  1  !
  67    0252  1
  68    0253  1  BUILTIN
  69    0254  1      SUBM;                        ! To support quadword subtraction
  70    0255  1
  71    0256  1  LITERAL
  72    0257  1      NICE_BUFLEN = 128;
  73    0258  1
  74    0259  1  !
  75    0260  1  !  TABLE OF CONTENTS:
  76    0261  1  !
  77    0262  1
  78    0263  1  FORWARD ROUTINE
  79    0264  1
  80    0265  1         PROCESS_SHOW,             ! Cover routine for common error handling of SHOW processing
  81    0266  1         SHOW_CIRCUIT,             ! Format circuit info
  82    0267  1         SHOW_SYSTEM;              ! Format info for a system ID message.
  83    0268  1
  84    0269  1  !
  85    0270  1  !  EXTERNAL REFERENCES:
  86    0271  1  !
  87    0272  1
  88    0273  1  EXTERNAL ROUTINE
  89    0274  1
  90    0275  1      ! Module CNFMAIN
  91    0276  1
  92    0277  1         CNF$EXIT,                 ! Clean up and exit
  93    0278  1         CNF$TRACE,                ! Log messages to log file
  94    0279  1         CNF$FREE_VM,              ! Free virtual memory
  95    0280  1         CNF$GET_ZVM,              ! Get zeroed virtual memory
  96    0281  1
  97    0282  1      ! Module CNFREQUES
  98    0283  1
  99    0284  1         CNF$LOCATE_CIR_BLK,       ! Locate circuit block from circuit name
 100    0285  1
 101    0286  1      ! Module CNFSEND
 102    0287  1
 103    0288  1         CNF$BUFR_NICE_MSG,        ! Buffer NICE response messages
 104    0289  1         CNF$BUFR_ERR_MSG;         ! Buffer NICE error responses
 105    0290  1
```

```
 106   0291  1 EXTERNAL
 107   0292  1
 108   0293  1         CNF$GQ_CIRSURLST : VECTOR [2];   ! List of circuits under surveillance
 109   0294  1
 110   0295  1 OWN
 111   0296  1         NICE_DONE_DSC :                  ! NICE 'DONE' message
 112   0297  1             BBLOCK [DSC$C_S_BLN] INITIAL
 113   0298  1             (
 114   0299  1                 1,
 115   0300  1                 UPLIT (
 116   0301  1                     BYTE (%X'80')
 117   0302  1                     )
 118   0303  1             ),
 119   0304  1
 120   0305  1         NICE_MORE_DSC :                  ! NICE 'MORE' message
 121   0306  1             BBLOCK [DSC$C_S_BLN] INITIAL
 122   0307  1             (
 123   0308  1                 4,
 124   0309  1                 UPLIT (
 125   0310  1                     BYTE (%X'02'),
 126   0311  1                     WORD (%X'FFFF'),
 127   0312  1                     BYTE (%X'00')
 128   0313  1                     )
 129   0314  1             );
```

CNF SHOW
V04-000

DECnet Ethernet Configurator Module   16-Sep-1984 02:05:37   VAX-11 Bliss-32 V4.0-742
CNF$PROCESS_SHOW   Search the data base and for 14-Sep-1984 12:49:54   [NICNF.SRC]CNFSHOW.B32;1

Page 4
     (3)

```
  131    0315  1  %SBTTL  'CNF$PROCESS_SHOW   Search the data base and format a response message'
  132    0316  1  GLOBAL ROUTINE CNF$PROCESS_SHOW (IRB, KNOWN, CIRCUITNAM_DSC, INFTYP) =
  133    0317  1
  134    0318  1  !++
  135    0319  1  ! FUNCTIONAL DESCRIPTION:
  136    0320  1  !
  137    0321  1  !   Shell routine to supply a common entrance and error exit to the
  138    0322  1  !   routine which builds the SHOW message.
  139    0323  1  !
  140    0324  1  ! FORMAL PARAMETERS:
  141    0325  1  !
  142    0326  1  !       irb             Interrupt request block, contains context for returning
  143    0327  1  !                       responses to connectee.
  144    0328  1  !
  145    0329  1  !       known           Was SHOW KNOWN CIRCUITS requested?
  146    0330  1  !
  147    0331  1  !       circuitnam_dsc  Descriptor of circuit name if SHOW was requested for
  148    0332  1  !                       a specific circuit.
  149    0333  1  !
  150    0334  1  !       inftyp          Code determining which information type was requested
  151    0335  1  !                       for the SHOW, either CHARACTERISTICS, SUMMARY or STATUS.
  152    0336  1  !
  153    0337  1  ! IMPLICIT INPUTS:
  154    0338  1  !       NONE
  155    0339  1  !
  156    0340  1  ! IMPLICIT OUTPUTS:
  157    0341  1  !       NONE
  158    0342  1  !
  159    0343  1  ! ROUTINE VALUE:
  160    0344  1  ! COMPLETION CODES:
  161    0345  1  !
  162    0346  1  !   Always success, errors are buffered for return to connectee.
  163    0347  1  !
  164    0348  1  ! SIDE EFFECTS:
  165    0349  1  !       NONE
  166    0350  1  !
  167    0351  1  !--
  168    0352  2    BEGIN
  169    0353  2    LOCAL
  170    0354  2        STATUS;
  171    0355  2
  172    0356  2    CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
  173    0357  2        $DESCRIPTOR ('CNF$PROCESS_SHOW'));
  174    0358  2
  175    0359  2    !
  176    0360  2    !   Send MORE message
  177    0361  2    !
  178    0362  2    EXECUTE (CNF$BUFR_NICE_MSG (.IRB, NICE_MORE_DSC, 0));
  179    0363  2
  180    0364  2    !
  181    0365  2    !   Request that the SHOW information be gathered, formatted and buffered.
  182    0366  2    !
  183    0367  2    STATUS = PROCESS_SHOW (.IRB, .KNOWN, .CIRCUITNAM_DSC, .INFTYP);
  184    0368  2    IF NOT .STATUS
  185    0369  2    THEN
  186    0370  2        CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_MPR, 0, .STATUS);
  187    0371  2
```

```
:  188      0372  2    !
:  189      0373  2    !  Send DONE message
:  190      0374  2    !
:  191      0375  2    EXECUTE (CNF$BUFR_NICE_MSG (.IRB, NICE_DONE_DSC, 0));
:  192      0376  2
:  193      0377  2    RETURN TRUE;
:  194      0378  1    END;                              ! Routine CNF$PROCESS_SHOW


                                                         .TITLE   CNFSHOW DECnet Ethernet Configurator Module
                                                         .IDENT   \V04-000\

                                                         .PSECT   $PLIT$,NOWRT,NOEXE,2

                                      80    00000 P.AAA:  .BYTE    -128                                              :
                                            00001         .BLKB    3
                                      02    00004 P.AAB:  .BYTE    2                                                 :
                                    FFFF    00005         .WORD    -1                                               :
                                      00    00007         .BYTE    0                                                :
                       45 43 41 52 54    00008 P.AAD:    .ASCII   \TRACE\                                          :
                                    0000D         .BLKB    3                                                        :
                              00000005    00010 P.AAC:    .LONG    5
                              00000000'   00014         .ADDRESS P.AAD
4F  4B  53  5F  53  53  45  43  4F  52  50  24  46  4E  43   00018 P.AAF:  .ASCII   \CNF$PROCESS_SHOW\              :
                                      57    00027
                              00000010    00028 P.AAE:    .LONG    16
                              00000000'   0002C         .ADDRESS P.AAF

                                                         .PSECT   $OWN$,NOEXE,2

                              00000001    00000 NICE_DONE_DSC:
                                                         .LONG    1
                              00000000'   00004         .ADDRESS P.AAA                                             :
                              00000004    00008 NICE_MORE_DSC:
                                                         .LONG    4
                              00000000'   0000C         .ADDRESS P.AAB                                             :

                                                         .EXTRN   CNF$EXIT, CNF$TRACE
                                                         .EXTRN   CNF$FREE_VM, CNF$GET_ZVM
                                                         .EXTRN   CNF$LOCATE_CIR_BLK
                                                         .EXTRN   CNF$BUFR_NICE_MSG
                                                         .EXTRN   CNF$BUFR_ERR_MSG
                                                         .EXTRN   CNF$GQ_CIRSURLST

                                                         .PSECT   $CODE$,NOWRT,2

                              0000 00000                 .ENTRY   CNF$PROCESS_SHOW, Save nothing            : 0316
                     0000'  CF  9F 00002                 PUSHAB   P.AAE                                     : 0357
                     0000'  CF  9F 00006                 PUSHAB   P.AAC                                     : 0356
                           01  DD 0000A                 PUSHL    #1
               0000G CF     03  FB 0000C                 CALLS    #3, CNF$TRACE
                           7E  D4 00011                 CLRL     -(SP)                                      : 0362
                     0000'  CF  9F 00013                 PUSHAB   NICE_MORE_DSC
                           04  AC  DD 00017              PUSHL    IRB
               0000G CF     03  FB 0001A                 CALLS    #3, CNF$BUFR_NICE_MSG
                     33     50  E9 0001F                 BLBC     STATUS, 2$
                     7E  0C  AC  7D 00022                 MOVQ     CIRCUITNAM_DSC, -(SP)                    : 0367
```

L 2

```
              7E      04   AC 7D 00026        MOVQ    IRB, -(SP)
        0000V CF      04   FB 0002A           CALLS   #4, PROCESS_SHOW
              0F           50 E8 0002F         BLBS    STATUS, 1$
                           50 DD 00032         PUSHL   STATUS
                           7E D4 00034         CLRL    -(SP)
              7E           05 CE 00036         MNEGL   #5, -(SP)
              04           AC DD 00039         PUSHL   IRB
        0000G CF      04   FB 0003C           CALLS   #4, CNF$BUFR_ERR_MSG
                      7E   D4 00041 1$:        CLRL    -(SP)
                    0000'  CF 9F 00043         PUSHAB  NICE_DONE_DSC
                      04   AC DD 00047         PUSHL   IRB
        0000G CF      03   FB 0004A           CALLS   #3, CNF$BUFR_NICE_MSG
              03           50 E9 0004F         BLBC    STATUS, 2$
              50           01 D0 00052         MOVL    #1, R0
                           04 00055 2$:        RET
```

```
; Routine Size: 86 bytes,    Routine Base: $CODE$ + 0000
```

```
196    0379  1  %SBTTL 'process_show    Search the data base and format a response message'
197    0380  1  ROUTINE PROCESS_SHOW (IRB, KNOWN, CIRCUITNAM_DSC, INFTYP) =
198    0381  1
199    0382  1  !++
200    0383  1  !   Locate requested circuit or dispatch for all known circuits to
201    0384  1  !   the routine which will format and buffer the SHOW response.
202    0385  1  !
203    0386  1  !       irb                Interrupt request block, contains context for returning
204    0387  1  !                          responses to connectee.
205    0388  1  !
206    0389  1  !       known              Was SHOW KNOWN CIRCUITS requested?
207    0390  1  !
208    0391  1  !       circuitnam_dsc     Descriptor of circuit name if SHOW was requested for
209    0392  1  !                          a specific circuit.
210    0393  1  !
211    0394  1  !       inftyp             Code determining which information type was requested
212    0395  1  !                          for the SHOW, either CHARACTERISTICS, SUMMARY or STATUS.
213    0396  1  !
214    0397  1  !   Always return success, any errors will be buffered for return to
215    0398  1  !   connectee.
216    0399  1  !--
217    0400  1
218    0401  2     BEGIN
219    0402  2     MAP
220    0403  2         CIRCUITNAM_DSC : REF BBLOCK;
221    0404  2
222    0405  2     LOCAL
223    0406  2         CIR : REF BBLOCK;
224    0407  2
225    0408  2
226    0409  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
227    0410  2         $DESCRIPTOR ('process_show'));
228    0411  2
229    0412  2     IF .KNOWN
230    0413  2     THEN
231    0414  2         !
232    0415  2         !   Format the data for all circuits
233    0416  2         !
234    0417  3         BEGIN
235    0418  3         CIR = .CNF$GQ_CIRSURLST;                    ! List of circuits under surveillance
236    0419  3         WHILE .CIR NEQ CNF$GQ_CIRSURLST DO         ! For the entire list of circuits
237    0420  4             BEGIN
238    0421  4             EXECUTE (SHOW_CIRCUIT (.IRB, .CIR, .INFTYP));
239    0422  4             CIR = .CIR [CIR$L_LINK];               ! Get next circuit in list
240    0423  4             END;                                  ! While traversing list of circuits
241    0424  3         END
242    0425  2     ELSE
243    0426  3         BEGIN
244    0427  3         !
245    0428  3         !   Locate the requested circuit and format the data for it.
246    0429  3         !
247    0430  3         IF CNF$LOCATE_CIR_BLK (.CIRCUITNAM_DSC, CIR)
248    0431  3         THEN
249    0432  4             EXECUTE (SHOW_CIRCUIT (.IRB, .CIR, .INFTYP))
250    0433  3         ELSE
251    0434  4             BEGIN                                    ! Oops, that circuit is not in the data base
252    0435  4             CNF$BUFR_ERR_MSG (.IRB, NMA$C_STS_IDE, NMA$C_ENT_CIR, 0, .CIRCUITNAM_DSC);
```

CNFSHOW
V04-000

DECnet Ethernet Configurator Module   16-Sep-1984 02:05:37   VAX-11 Bliss-32 V4.0-742   Page 8
process_show   Search the data base and format 14-Sep-1984 12:49:54   [NICNF.SRC]CNFSHOW.B32;1   (4)

```
;   253        0436  4                   RETURN TRUE;
;   254        0437  3                   END;
;   255        0438  3
;   256        0439  2           END;
;   257        0440  2
;   258        0441  2       RETURN TRUE;
;   259        0442  1       END;                            ! Routine process_show


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                      45 43 41 52 54  00030 P.AAH:  .ASCII  \TRACE\
                                      00035          .BLKB   3
                            00000005  00038 P.AAG:  .LONG   5
                            00000000' 0003C          .ADDRESS P.AAH
      77 6F 68 73 5F 73 73 65 63 6F 72 70  00040 P.AAJ:  .ASCII  \process_show\
                            0000000C  0004C P.AAI:  .LONG   12
                            00000000' 00050          .ADDRESS P.AAJ


                                                .PSECT  $CODE$,NOWRT,2

                            0000  00000 PROCESS_SHOW:
                                                       .WORD   Save nothing
                   5E            04 C2 00002          SUBL2   #4, SP
                        0000' CF 9F 00005          PUSHAB  P.AAI
                        0000' CF 9F 00009          PUSHAB  P.AAG
                              01 DD 0000D          PUSHL   #1
              0000G CF       03 FB 0000F          CALLS   #3, CNF$TRACE
                    24       08 AC E9 00014          BLBC    KNOWN, 2$
                    6E 0000G CF D0 00018          MOVL    CNF$GQ_CIRSURLST, CIR
                    50 0000G CF 9E 0001D 1$:      MOVAB   CNF$GQ_CIRSURLST, R0
                    50          6E D1 00022          CMPL    CIR, R0
                              45 13 00025          BEQL    4$
                           10 AC DD 00027          PUSHL   INFTYP
                           04 AE DD 0002A          PUSHL   CIR
                           04 AC DD 0002D          PUSHL   IRB
              0000V CF       03 FB 00030          CALLS   #3, SHOW_CIRCUIT
                    37          50 E9 00035          BLBC    STATUS, 5$
                              9E DD 00038          PUSHL   @CIR
                              E1 11 0003A          BRB     1$
                    5E          DD 0003C 2$:      PUSHL   SP
                    0C       AC DD 0003E          PUSHL   CIRCUITNAM_DSC
              0000G CF       02 FB 00041          CALLS   #2, CNF$LOCATE_CIR_BLK
                    12          50 E9 00046          BLBC    R0, 3$
                           10 AC DD 00049          PUSHL   INFTYP
                           04 AE DD 0004C          PUSHL   CIR
                           04 AC DD 0004F          PUSHL   IRB
              0000V CF       03 FB 00052          CALLS   #3, SHOW_CIRCUIT
                    12          50 E8 00057          BLBS    STATUS, 7$
                              04 0005A          RET
                    0C       AC DD 0005B 3$:      PUSHL   CIRCUITNAM_DSC
                    7E          03 7D 0005E          MOVQ    #3, -(SP)
                    7E          09 CE 00061          MNEGL   #9, -(SP)
                    04       AC DD 00064          PUSHL   IRB
```

```
;
;
;
;
;


; 0380
; 0410
; 0409

; 0412
; 0418
; 0419

; 0421


; 0422
; 0419
; 0430

; 0432


; 0435
```

```
                    0000G  CF            05 FB 00067        CALLS    #5, CNF$BUFR_ERR_MSG
                           50            01 DO 0006C 4$:    MOVL     #1, R0
                                         04 0006F 5$:       RET
```
                                                                                                              : 0441
                                                                                                              : 0442

; Routine Size: 112 bytes,   Routine Base: $CODE$ + 0056

```
 261   0443  1  %SBTTL 'show_circuit   Format all systems for circuit'
 262   0444  1  ROUTINE SHOW_CIRCUIT (IRB, CIR, INFTYP) =
 263   0445  1
 264   0446  1  !++
 265   0447  1  !   Build the NICE for the SHOW response message and buffer it for
 266   0448  1  !   transmission to the connectee.
 267   0449  1  !
 268   0450  1  !       irb             Interrupt request block, contains context for returning
 269   0451  1  !                       responses to connectee.
 270   0452  1  !
 271   0453  1  !       cir             Address of Circuit control block of circuit SHOW
 272   0454  1  !                       was requested for.
 273   0455  1  !
 274   0456  1  !       inftyp          Code determining which information type was requested
 275   0457  1  !                       for the SHOW, either CHARACTERISTICS, SUMMARY or STATUS.
 276   0458  1  !
 277   0459  1  !   Always return success, any errors will be buffered for return to
 278   0460  1  !   connectee.
 279   0461  1  !--
 280   0462  1
 281   0463  2      BEGIN
 282   0464  2      MAP
 283   0465  2          CIR : REF BBLOCK;
 284   0466  2
 285   0467  2      LOCAL
 286   0468  2          CURRENT_TIMBUF : BBLOCK [8],              ! Buffer to obtain the current system time
 287   0469  2          DELTA_TIMBUF : BBLOCK [8],                ! Buffer to calculate the time difference between the curren
 288   0470  2                                                   !   the time surveillance began on the circuit.
 289   0471  2          NICE : REF BBLOCK,                       ! Pointer into the buffer where the NICE message is being bu
 290   0472  2          NICE_BUFDSC : BBLOCK [DSC$C_S_BLN],       ! Descriptor of NICE message buffer
 291   0473  2          NICE_TMPDSC : BBLOCK [DSC$C_S_BLN];       ! Descriptor of NICE Template buffer
 292   0474  2          SID : REF BBLOCK,                         ! Pointer to a system ID message
 293   0475  2          TIMBUF : VECTOR [7, WORD];                ! Buffer for converting binary time format to ASCII for NICE
 294   0476  2
 295   0477  2      BIND
 296   0478  2          CONF = UPLIT (%ASCIC 'CONFIGURATOR') : VECTOR [,BYTE];  ! Module name to place into NICE return
 297   0479  2
 298   0480  2
 299   0481  2      CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
 300   0482  2          $DESCRIPTOR ('show_circuit'));
 301   0483  2
 302   0484  2      !
 303   0485  2      !   Zero the descriptor which will locate the buffer where the NICE response
 304   0486  2      !   will be built, allocate the buffer, and initialize buffer pointer.
 305   0487  2      !
 306   0488  2      CH$FILL (0, DSC$C_S_BLN, NICE_TMPDSC);
 307   0489  2      EXECUTE (CNF$GET_ZVM (%REF (NICE_BUFLEN), NICE_TMPDSC [DSC$A_POINTER]));
 308   0490  2      NICE = .NICE_TMPDSC [DSC$A_POINTER];
 309   0491  2
 310   0492  2      !
 311   0493  2      !   Place Error status
 312   0494  2      !
 313   0495  2      !       1 byte          return code
 314   0496  2      !       2 bytes         error detail
 315   0497  2      !       1 byte          length of error message
 316   0498  2      !
 317   0499  2      (.NICE) <0, 8> = %X'01';                      ! Return code SUCCESS
```

```
 318   0500  2           (.NICE) <8, 16> = %X'FFFF';                          ! Error detail, SUCCESS
 319   0501  2           (.NICE) <24, 8> = %X'00';                            ! Error text length
 320   0502  2
 321   0503  2           !
 322   0504  2           !    Copy over the module entity, CONFIGURATOR
 323   0505  2           !
 324   0506  2           !        1 byte      Length of CONFIGURATOR string
 325   0507  2           !        12 bytes    CONFIGURATOR string
 326   0508  2           !
 327   0509  2           (.NICE) <32, 8> = .CONF [0];                         ! Length of CONFIGURATOR string
 328   0510  2           NICE = .NICE + 5;                                    ! Set pointer to beginning of circuit name
 329   0511  2           CH$MOVE (.CONF [0], CONF [1], .NICE);
 330   0512  2           NICE_TMPDSC [DSC$W_LENGTH] = 5 + .CONF [0];
 331   0513  2           NICE = .NICE + .CONF [0];                            ! Point to free space in buffer after
 332   0514  2                                                               !  the circuit name which was just copied in
 333   0515  2
 334   0516  2           !
 335   0517  2           !    Copy over Circuit name entity
 336   0518  2           !
 337   0519  2           !        2 bytes     Circuit entity ID
 338   0520  2           !        1 byte      Parameter type = ASCII
 339   0521  2           !        1 byte      Length of circuit name
 340   0522  2           !        n bytes     Circuit name
 341   0523  2           !
 342   0524  2           (.NICE) <0, 16> = NMA$C_PCCN_CIR;
 343   0525  2           (.NICE) <16, 8> = NMA$C_PTY_AI;                      ! Type = ASCII
 344   0526  2           (.NICE) <24, 8> = .CIR [CIR$W_CIRNAMLEN];            ! Length of Circuit name
 345   0527  2           NICE = .NICE + 4;                                    ! Set pointer to beginning of circuit name
 346   0528  2           CH$MOVE (.CIR [CIR$W_CIRNAMLEN], CIR [CIR$T_CIRNAM], .NICE);
 347   0529  2           NICE = .NICE + .CIR [CIR$W_CIRNAMLEN];               ! Point to free space in buffer after
 348   0530  2                                                               !  the circuit name which was just copied in
 349   0531  2           NICE_TMPDSC [DSC$W_LENGTH] = .NICE_TMPDSC [DSC$W_LENGTH] + 4 + .CIR [CIR$W_CIRNAMLEN];
 350   0532  2
 351   0533  2           !
 352   0534  2           !    Place in Surveillance parameter
 353   0535  2           !    as a coded value
 354   0536  2           !
 355   0537  2           !        2 bytes     Surviellance parameter ID
 356   0538  2           !        1 byte      Surveillance type = coded byte
 357   0539  2           !        1 byte      Surveillance value
 358   0540  2           !
 359   0541  2           (.NICE) <0, 16> = NMA$C_PCCN_SUR;
 360   0542  3           BEGIN
 361   0543  3           BIND
 362   0544  3               TYPE = .NICE + 2 : BBLOCK;
 363   0545  3           TYPE [NMA$V_PTY_COD] = TRUE;                         ! Surveillance is returned as a coded value
 364   0546  3           TYPE [NMA$V_PTY_CLE] = 1;                            ! The coded value is 1 byte in length
 365   0547  3           END;
 366   0548  2           (.NICE) <24, 8> = .CIR [CIR$B_SURVEIL];
 367   0549  2           NICE = .NICE + 4;                                    ! Set pointer to end of buffer where Elapsed Time will be pl
 368   0550  2
 369   0551  2           !
 370   0552  2           !    Place in Elapsed Time parameter
 371   0553  2           !    as a coded multiple
 372   0554  2           !
 373   0555  2           !        2 bytes     Elasped Time parameter ID
 374   0556  2           !        1 byte      Elasped time type = coded multiple of 3 fields
```

E 3

```
375    0557  2  |          1 byte      hours type = unsigned decimal word
376    0558  2  |          2 bytes     hours value
377    0559  2  |          1 byte      minutes type = unsigned decimal byte
378    0560  2  |          1 byte      minutes value
379    0561  2  |          1 byte      seconds type = unsigned decimal byte
380    0562  2  |          1 byte      seconds value
381    0563  2
382    0564  2          (.NICE) <0, 16> = NMASC_PCCN_ELT;                        ! Set parameter ID
383    0565  3              BEGIN
384    0566  3              BIND
385    0567  3                  CODMUL_TYP = .NICE + 2 : BBLOCK,
386    0568  3                  HR_TYP = .NICE + 3 : BBLOCK,
387    0569  3                  MIN_TYP = .NICE + 6 : BBLOCK,
388    0570  3                  SEC_TYP = .NICE + 8 : BBLOCK;
389    0571  3
390    0572  3              CODMUL_TYP [NMASV_PTY_COD] = TRUE;                   ! Elapsed Time is returned as a coded
391    0573  3              CODMUL_TYP [NMASV_PTY_MUL] = TRUE;                   !  multiple.
392    0574  3              CODMUL_TYP [NMASV_PTY_CLE] = 3;                      ! There are three fields in the coded multiple
393    0575  3
394    0576  3              |
395    0577  3              |   Get the current system time, subtract
396    0578  3              |   Time of Surveillance start from Current time
397    0579  3              |   to get negative Delta time
398    0580  3              |
399    0581  3              EXECUTE ($GETTIM (TIMADR = CURRENT_TIMBUF));
400    0582  3              SUBM (2, CIR [CIRSQ_ELAPSDTIM], CURRENT_TIMBUF, DELTA_TIMBUF);
401    0583  3              EXECUTE ($NUMTIM (TIMBUF = TIMBUF, TIMADR = DELTA_TIMBUF));
402    0584  3
403    0585  3              HR_TYP [NMASV_PTY_NTY] = NMASC_NTY_DU;               ! Unsigned decimal
404    0586  3              HR_TYP [NMASV_PTY_NLE] = 2;                          !  word.
405    0587  3              (.NICE) <32, 16> = .TIMBUF [3];                      ! Hours
406    0588  3
407    0589  3              MIN_TYP [NMASV_PTY_NTY] = NMASC_NTY_DU;              ! Unsigned decimal
408    0590  3              MIN_TYP [NMASV_PTY_NLE] = 1;                         !  byte.
409    0591  3              (.NICE) <56, 8> = .TIMBUF [4];                       ! Minutes
410    0592  3
411    0593  3              SEC_TYP [NMASV_PTY_NTY] = NMASC_NTY_DU;              ! Unsigned decimal
412    0594  3              SEC_TYP [NMASV_PTY_NLE] = 1;                         !  byte.
413    0595  3              (.NICE) <72, 8> = .TIMBUF [5];                       ! Seconds
414    0596  3              END;
415    0597  2
416    0598  2          NICE_TMPDSC [DSC$W_LENGTH] = 14 + .NICE_TMPDSC [DSC$W_LENGTH];
417    0599  2
418    0600  2          SID = .CIR [CIR$L_SIDFLINK];                            ! Point to first System ID
419    0601  2
420    0602  2          IF (.SID EQL CIR [CIR$L_SIDFLINK]) OR                   ! There are no ID's collected for this circuit
421    0603  2              ((.INFTYP NEQ NMASC_OPINF_STA) AND                   ! or Summary requested
422    0604  2               (.INFTYP NEQ NMASC_OPINF_CHA))
423    0605  2          THEN
424    0606  2              |
425    0607  2              |   Print only circuit info, not system ID's, since either
426    0608  2              |   there are no ID's collected, or a SHOW SUMMARY was requested.
427    0609  2              |
428    0610  3              BEGIN
429    0611  3              CNF$BUFR NICE_MSG (.IRB, NICE_TMPDSC, NICE_BUFLEN);
430    0612  3              RETURN TRUE;
431    0613  3              END
```

```
  432   0614   3          ELSE
  433   0615   3
  434   0616   3               !
  435   0617   3               !    Traverse the list of system ID's and format a NICE response
  436   0618   3               !    for each one.  Each one will be appended to a repeat of the
  437   0619   3               !    circuit info already gathered.
  438   0620   3               !
  439   0621   3               BEGIN
  440   0622   3               WHILE .SID NEQ CIR [CIR$L_SIDFLINK] DO           ! For all the System ID's
  441   0623   4                   BEGIN
  442   0624   4
  443   0625   4                   !
  444   0626   4                   !    Zero the descriptor and allocate a clean buffer for the NICE
  445   0627   4                   !    response message.  Then copy the message already built for the
  446   0628   4                   !    circuit info as the start of the message to which the system ID
  447   0629   4                   !    info will be appended.
  448   0630   4                   !
  449   0631   4                   CH$FILL (0, DSC$C_S_BLN, NICE_BUFDSC);
  450   0632   4                   EXECUTE (CNF$GET_VM (%REF (NICE_BUFLEN), NICE_BUFDSC [DSC$A_POINTER]));
  451   0633   4                   CH$MOVE (.NICE_TMPDSC [DSC$W_LENGTH], .NICE_TMPDSC [DSC$A_POINTER],
  452   0634   4                       .NICE_BUFDSC [DSC$A_POINTER]);
  453   0635   4                   NICE_BUFDSC [DSC$W_LENGTH] = .NICE_TMPDSC [DSC$W_LENGTH];
  454   0636   4
  455   0637   4                   !
  456   0638   4                   !    Append the system ID info to the NICE response, and
  457   0639   4                   !    buffer the message for later transmission.
  458   0640   4                   !    Then follow list pointer to next system ID.
  459   0641   4                   !
  460   0642   4                   SHOW_SYSTEM (.SID, NICE_BUFDSC);
  461   0643   4                   CNF$BUFR_NICE_MSG (.IRB, NICE_BUFDSC, NICE_BUFLEN);
  462   0644   4                   SID = .SID [SID$L_LINK];
  463   0645   4                   END;                            ! While processing all system ID's for the circuit
  464   0646   3
  465   0647   3               !
  466   0648   3               !    Return the buffer which we used to build the circuit info part
  467   0649   3               !    of the response.
  468   0650   3               !
  469   0651   3               EXECUTE (CNF$FREE_VM (%REF (NICE_BUFLEN), NICE_TMPDSC [DSC$A_POINTER]));
  470   0652   2               END;                                ! There are system ID's for this circuit
  471   0653   2
  472   0654   2           RETURN TRUE;
  473   0653   1           END;                                    ! Routine show_circuit
```

```
                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

00  00  52  4F  54  41  52  55  47  49  46  4E  4F  43  0C  00054 P.AAK:   .ASCII  <12>\CONFIGURATOR\<0><0><0>
                                                      00  00063
                                      45  43  41  52  54  00064 P.AAM:   .ASCII  \TRACE\
                                                          00069          .BLKB   3
                                              00000005  0006C P.AAL:   .LONG   5
                                              00000000' 00070          .ADDRESS P.AAM
              74  69  75  63  72  69  63  5F  77  6F  68  73  00074 P.AAO:   .ASCII  \show_circuit\
                                              0000000C  00080 P.AAN:   .LONG   12
                                              00000000' 00084          .ADDRESS P.AAO

                                              CONF=       P.AAK
                                                      .EXTRN  SYS$GETTIM, SYS$NUMTIM
```

CNFSHOW
V04-000

DECnet Ethernet Configurator Module
show_circuit   Format all systems for circuit

G 3
16-Sep-1984 02:05:37   VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:49:54   [NICNF.SRC]CNFSHOW.B32;1

Page 14
(5)

```
                                                        .PSECT  $CODE$,NOWRT,2

                                    00FC 00000 SHOW_CIRCUIT:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7                       0444
                        5E          34  C2 00002        SUBL2   #52, SP
                            0000'   CF  9F 00005        PUSHAB  P.AAN                                       0482
                            0000'   CF  9F 00009        PUSHAB  P.AAL                                       0481
                                    01  DD 0000D        PUSHL   #1
                0000G   CF          03  FB 0000F        CALLS   #3, CNF$TRACE
     08         00      6E          00  2C 00014        MOVC5   #0, (SP), #0, #8, NICE_TMPDSC               0488
                            14      AE     00019
                            18      AE  9F 0001B        PUSHAB  NICE_TMPDSC+4                               0489
                04  AE     80      8F  9A 0001E        MOVZBL  #128, 4(SP)
                04      AE  9F 00023        PUSHAB  4(SP)
                0000G   CF          02  FB 00026        CALLS   #2, CNF$GET_ZVM
                        50  E9 0002B        BLBC    STATUS, 1$
                        79
                        56          18  AE  D0 0002E    MOVL    NICE_TMPDSC+4, NICE                         0490
                        86          01  90 00032        MOVB    #1, (NICE)+                                 0499
                        86          01  AE 00035        MNEGW   #1, (NICE)+                                 0500
                        86          94 00038        CLRB    (NICE)+                                     0501
                        57  0000'   CF  9A 0003A        MOVZBL  CONF, R7                                    0509
                        96          57  90 0003F        MOVB    R7, (NICE)+
                14  66  0000'   CF          57  28 00042    MOVC3   R7, CONF+1, (NICE)                          0511
                    AE          57          05  A1 00048    ADDW3   #5, R7, NICE_TMPDSC                         0512
                        56          57  C0 0004D        ADDL2   R7, NICE                                    0513
                        86          64  8F  9B 00050    MOVZBW  #100, (NICE)+                               0524
                        86          40  8F  90 00055    MOVB    #64, (NICE)+                                0525
                        57          08  A6  D0 00058    MOVL    CIR, R7                                     0526
                        86          16  A7  90 0005C    MOVB    22(R7), (NICE)+                             0528
                66  18  A7          16  A7  28 00060    MOVC3   22(R7), 24(R7), (NICE)                      0529
                        50          16  A7  32 00066    CVTWL   22(R7), R0
                        56          50  C0 0006A        ADDL2   R0, NICE
                        50  14  AE          3C 0006D    MOVZWL  NICE_TMPDSC, R0                             0531
                        51          16  A7  32 00071    CVTWL   22(R7), R1
                        50          51  C0 00075        ADDL2   R1, R0
                14  AE          50          04  A1 00078    ADDW3   #4, R0, NICE_TMPDSC
                        86          6E  8F  9B 0007D    MOVZBW  #110, (NICE)+                               0541
                        66          80  8F  88 00081    BISB2   #128, (NICE)                               0545
            B6  06          00          01  F0 00085    INSV    #1, #0, #6, (NICE)+                         0546
                        86          0A  A7  90 0008A    MOVB    10(R7), (NICE)+                             0548
                        66          6F  8F  9B 0008E    MOVZBW  #111, (NICE)                                0564
                02  A6          CO  8F  88 00092    BISB2   #192, 2(NICE)                              0573
    02  A6      06          00          03  F0 00097    INSV    #3, #0, #6, 2(NICE)                         0574
                        2C          AE  9F 0009D        PUSHAB  CURRENT_TIMBUF                             0581
            000000000G  00          01  FB 000A0    CALLS   #1, SYS$GETTIM
                        50  E9 000A7 1$:   BLBC    STATUS, 2$
                        1E
                24  AE          2C  AE          30  A7  C3 000AA   SUBL3  48(R7), CURRENT_TIMBUF, DELTA_TIMBUF   0582
                        28  AE          30  AE  D0 000B1   MOVL   CURRENT_TIMBUF, DELTA_TIMBUF
                        28  AE          34  A7  D9 000B6   SBWC   52(R7), DELTA_TIMBUF
                        24          AE  9F 000BB        PUSHAB  DELTA_TIMBUF                               0583
                        08          AE  9F 000BE        PUSHAB  TIMBUF
            000000000G  00          02  FB 000C1    CALLS   #2, SYS$NUMTIM
                        50  E9 000C8 2$:   BLBC    STATUS, 5$
                        78
                03  A6          30  8A 000CB        BICB2   #48, 3(NICE)                               0585
    03  A6      04          02  F0 000CF        INSV    #2, #0, #4, 3(NICE)                         0586
                04  A6          0A          AE  B0 000D5    MOVW   TIMBUF+6, 4(NICE)                          0587
```

```
                              06  A6      30  8A  000DA        BICB2    #48, 6(NICE)                              0589
     06  A6            04      00          01  FO  000DE        INSV     #1, #0, #4, 6(NICE)                       0590
                              07  A6   0C  AE  90  000E4        MOVB     TIMBUF+8, 7(NICE)                         0591
                              08  A6      30  8A  000E9        BICB2    #48, 8(NICE)                              0593
     08  A6            04      00          01  FO  000ED        INSV     #1, #0, #4, 8(NICE)                       0594
                              09  A6   0E  AE  90  000F3        MOVB     TIMBUF+10, 9(NICE)                        0595
                              14  AE      0E  A0  000F8        ADDW2    #14, NICE_TMPDSC                           0598
                                  56  40  A7  DO  000FC        MOVL     64(R7), SID                               0600
                                  50  40  A7  9E  00100        MOVAB    64(R7), RO                                0602
                                  50      56  D1  00104        CMPL     SID, RO
                                          0C  13  00107        BEQL     3$
                                  01  0C  AC  D1  00109        CMPL     INFTYP, #1                                0603
                                          17  13  0010D        BEQL     4$
                                  02  0C  AC  D1  0010F        CMPL     INFTYP, #2                                0604
                                          11  13  00113        BEQL     4$
                              7E  80  8F  9A  00115  3$:       MOVZBL   #128, -(SP)                               0611
                                  18  AE  9F  00119            PUSHAB   NICE_TMPDSC
                                  04  AC  DD  0011C            PUSHL    IRB
                      0000G  CF  03  FB  0011F            CALLS    #3, CNF$BUFR_NICE_MSG
                                  60  11  00124            BRB      7$                                            0612
                              50  40  A7  9E  00126  4$:       MOVAB    64(R7), RO                                0622
                                  50      56  D1  0012A        CMPL     SID, RO
                                          44  13  0012D        BEQL     6$
     08          00      6E  00  2C  0012F        MOVC5    #0, (SP), #0, #8, NICE_BUFDSC             0630
                              1C  AE      00134
                              20  AE  9F  00136            PUSHAB   NICE_BUFDSC+4                              0631
                          04  AE  80  8F  9A  00139        MOVZBL   #128, 4(SP)
                              04  AE  9F  0013E            PUSHAB   4(SP)
                      0000G  CF  02  FB  00141            CALLS    #2, CNF$GET_ZVM
                                  40      50  E9  00146  5$:       BLBC     STATUS, 8$
     20  BE      18  BE  14  AE  28  00149        MOVC3    NICE_TMPDSC, @NICE_TMPDSC+4, @NICE_BUFDSC+4   0633
                  1C  AE  14  AE  B0  00150        MOVW     NICE_TMPDSC, NICE_BUFDSC                   0634
                              1C  AE  9F  00155            PUSHAB   NICE_BUFDSC                               0641
                                  56  DD  00158            PUSHL    SID
                      0000V  CF  02  FB  0015A            CALLS    #2, SHOW_SYSTEM
                              7E  80  8F  9A  0015F        MOVZBL   #128, -(SP)                               0642
                              20  AE  9F  00163            PUSHAB   NICE_BUFDSC
                                  04  AC  DD  00166            PUSHL    IRB
                      0000G  CF  03  FB  00169            CALLS    #3, CNF$BUFR_NICE_MSG
                                  66  DO  0016E            MOVL     (SID), SID                                0643
                                  B3  11  00171            BRB      4$                                        0622
                              18  AE  9F  00173  6$:       PUSHAB   NICE_TMPDSC+4                             0649
                          04  AE  80  8F  9A  00176        MOVZBL   #128, 4(SP)
                              04  AE  9F  0017B            PUSHAB   4(SP)
                      0000G  CF  02  FB  0017E            CALLS    #2, CNF$FREE_VM
                                  50  E9  00183            BLBC     STATUS, 8$
                                  01  DO  00186  7$:       MOVL     #1, RO                                    0652
                                  04  00189  8$:       RET                                                     0653
```

; Routine Size: 394 bytes,   Routine Base: $CODE$ + 00C6

CNFSHOW
V04-000

DECnet Ethernet Configurator Module
show_system   Format System ID info

| 3
16-Sep-1984 02:05:37
14-Sep-1984 12:49:54

VAX-11 Bliss-32 V4.0-742
[NICNF.SRC]CNFSHOW.B32;1

Page 16
(6)

```
473   0654  1   %SBTTL 'show_system   Format System ID info'
474   0655  1   ROUTINE SHOW_SYSTEM (SID, NICEBUF) =
475   0656  1
476   0657  1   !++
477   0658  1   !   Format the information in the system ID message stored in
478   0659  1   !   SID and build a NICE message which will be appended to the
479   0660  1   !   NICE message for the circuit which is in NICEBUF.
480   0661  1   !
481   0662  1   !       sid       Pointer to buffer containing a system ID message
482   0663  1   !
483   0664  1   !       nicebuf   Descriptor of buffer containing circuit NICE message
484   0665  1   !
485   0666  1   !   Always return success.  There is no error checking.
486   0667  1   !
487   0668  1   !--
488   0669  2   BEGIN
489   0670      MAP
490   0671          NICEBUF : REF BBLOCK,
491   0672          SID : REF BBLOCK;
492   0673      LOCAL
493   0674          NICE : REF BBLOCK,
494   0675          TIMBUF : VECTOR [7, WORD];
495   0676
496   0677  2   CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
497   0678  2       $DESCRIPTOR ('show_system'));
498   0679  2
499   0680  2   NICE = .NICEBUF [DSC$A_POINTER] + .NICEBUF [DSC$W_LENGTH];
500   0681  2
501   0682      !
502   0683      !   Place in Physical Address parameter
503   0684      !   as a Hex Image 6
504   0685      !
505   0686      !       2 bytes      Physical Address parameter ID
506   0687      !       1 byte       Physical Address type = Hex Image (HI-6)
507   0688      !       6 bytes      Physical Address value
508   0689      !
509   0690  2   (.NICE) <0, 16> = NMA$C_PCCN_PHA;
510   0691      BEGIN
511   0692      BIND
512   0693          TYPE = .NICE + 2 : BBLOCK;
513   0694      TYPE [NMA$V_PTY_NTY] = NMA$C_NTY_H;     ! returned as a Hex
514   0695      TYPE [NMA$V_PTY_NLE] = NMA$C_NLE_IMAGE; !   image.
515   0696      END;
516   0697  2   (.NICE) <24,8> = SID$C_ADRLEN;
517   0698  2   CH$MOVE (SID$C_ADRLEN, SID [SID$T_CURADR], (.NICE + 4) );
518   0699  2   NICE = .NICE + 4 + SID$C_ADRLEN;                ! Set pointer to end of buffer where next parameter will be
```

```
520   0700   2                 Place in Last Report parameter
521   0701   2                 as a coded multiple
522   0702   2
523   0703   2
524   0704   2                     2 bytes       Last Report parameter ID
525   0705   2                     1 byte        Last Report type = coded multiple of 5 fields
526   0706   2                     1 byte        Day type = unsigned decimal byte
527   0707   2                     1 byte        Day value
528   0708   2                     1 byte        Month type = Coded byte
529   0709   2                     1 byte        Month coded value
530   0710   2                     1 byte        hour type = unsigned decimal byte
531   0711   2                     1 byte        hour value
532   0712   2                     1 byte        minutes type = unsigned decimal byte
533   0713   2                     1 byte        minutes value
534   0714   2                     1 byte        seconds type = unsigned decimal byte
535   0715   2                     1 byte        seconds value
536   0716   2
537   0717   2             (.NICE) <0, 16> = NMASC_PCCN_LRP;
538   0718   2                 BEGIN
539   0719   2                 BIND
540   0720   3                     CODMUL_TYP = .NICE + 2 : BBLOCK,
541   0721   3                     DAY_TYP = .NICE + 3 : BBLOCK,
542   0722   3                     MON_TYP = .NICE + 5 : BBLOCK,
543   0723   3                     HR_TYP = .NICE + 7 : BBLOCK,
544   0724   3                     MIN_TYP = .NICE + 9 : BBLOCK,
545   0725   3                     SEC_TYP = .NICE + 11 : BBLOCK;
546   0726   3
547   0727   3                 CODMUL_TYP [NMASV_PTY_COD] = TRUE;        ! Last Report is returned as a coded
548   0728   3                 CODMUL_TYP [NMASV_PTY_MUL] = TRUE;        !  multiple.
549   0729   3                 CODMUL_TYP [NMASV_PTY_CLE] = 5;           ! There are five fields in the coded multiple
550   0730   3
551   0731   3                 EXECUTE ($NUMTIM (TIMBUF = TIMBUF, TIMADR = SID [SID$Q_LSTREPORT]) );
552   0732   3
553   0733   3                 DAY_TYP [NMASV_PTY_NTY] = NMASC_NTY_DU;  ! Unsigned decimal
554   0734   3                 DAY_TYP [NMASV_PTY_NLE] = 1;             !  byte.
555   0735   3                 (.NICE) <32, 8> = .TIMBUF [2];           ! Day
556   0736   3
557   0737   3                 MON_TYP [NMASV_PTY_COD] = TRUE;          ! Month is returned as a coded value
558   0738   3                 MON_TYP [NMASV_PTY_CLE] = 1;             !  contained in 1 byte.
559   0739   3                 (.NICE) <48, 8> = .TIMBUF [1];           ! Month
560   0740   3
561   0741   3                 HR_TYP [NMASV_PTY_NTY] = NMASC_NTY_DU;   ! Unsigned decimal
562   0742   3                 HR_TYP [NMASV_PTY_NLE] = 1;              !  byte.
563   0743   3                 (.NICE) <64, 8> = .TIMBUF [3];           ! Hour
564   0744   3
565   0745   3                 MIN_TYP [NMASV_PTY_NTY] = NMASC_NTY_DU;  ! Unsigned decimal
566   0746   3                 MIN_TYP [NMASV_PTY_NLE] = 1;             !  byte.
567   0747   3                 (.NICE) <80, 8> = .TIMBUF [4];           ! Minute
568   0748   3
569   0749   3                 SEC_TYP [NMASV_PTY_NTY] = NMASC_NTY_DU;  ! Unsigned decimal
570   0750   3                 SEC_TYP [NMASV_PTY_NLE] = 1;             !  byte.
571   0751   3                 (.NICE) <96, 8> = .TIMBUF [5];           ! Second
572   0752   3                 END;
573   0753   2             NICE = .NICE + 13;
574   0754   2
```

```
576  0755  2              Place in Maintenance Version parameter
577  0756  2              as a coded multiple
578  0757  2
579  0758  2
580  0759  2                  2 bytes      Maintenance Version parameter ID
581  0760  2                  1 byte       Maintenance Version type = coded multiple of 3 fields
582  0761  2                  1 byte       Version Number type = unsigned decimal byte
583  0762  2                  1 byte       Version Number value
584  0763  2                  1 byte       ECO number type = unsigned decimal byte
585  0764  2                  1 byte       ECO number value
586  0765  2                  1 byte       User ECO number type = unsigned decimal byte
587  0766  2                  1 byte       User ECO value
588  0767  2
589  0768  2          (.NICE) <0, 16> = NMASC_PCCN_MVR;
590  0769  2              BEGIN
591  0770  3              BIND
592  0771  3                  CODMUL_TYP = .NICE + 2 : BBLOCK,
593  0772  3                  VN_TYP = .NICE + 3 : BBLOCK,
594  0773  3                  EN_TYP = .NICE + 5 : BBLOCK,
595  0774  3                  UEN_TYP = .NICE + 7 : BBLOCK;
596  0775  3
597  0776  3              CODMUL_TYP [NMASV_PTY_COD] = TRUE;              ! Maintenance Version is returned as a coded
598  0777  3              CODMUL_TYP [NMASV_PTY_MUL] = TRUE;              !  multiple.
599  0778  3              CODMUL_TYP [NMASV_PTY_CLE] = 3;                 ! There are three fields in the coded multiple
600  0779  3
601  0780  3              VN_TYP [NMASV_PTY_NTY] = NMASC_NTY_DU;          ! Unsigned decimal
602  0781  3              VN_TYP [NMASV_PTY_NLE] = 1;                     !  byte.
603  0782  3              (.NICE) <32, 8> = .SID [SID$B_MOPVER];          ! MOP version
604  0783  3
605  0784  3              EN_TYP [NMASV_PTY_NTY] = NMASC_NTY_DU;          ! Unsigned decimal
606  0785  3              EN_TYP [NMASV_PTY_NLE] = 1;                     !  byte.
607  0786  3              (.NICE) <48, 8> = .SID [SID$B_MOPECO];          ! MOP ECO
608  0787  3
609  0788  3              UEN_TYP [NMASV_PTY_NTY] = NMASC_NTY_DU;         ! Unsigned decimal
610  0789  3              UEN_TYP [NMASV_PTY_NLE] = 1;                    !  byte.
611  0790  3              (.NICE) <64, 8> = .SID [SID$B_MOPUSRECO];
612  0791  2              END;
613  0792  2
614  0793  2          NICE = .NICE + 9;
615  0794  2
```

```
  617    0795   2              !       Place in Functions parameter
  618    0796   2              !       as a coded multiple
  619    0797   2              !
  620    0798   2              !           2 bytes     Functions parameter ID
  621    0799   2              !           1 byte      Functions type = coded multiple of up to 16 fields
  622    0800   2              !           n bytes     Function type = Coded byte
  623    0801   2              !
  624    0802   2              !           up to 16 functions permitted
  625    0803   2              !
  626    0804   2              !
  627    0805   2
  628    0806   2              IF .SID [SID$B_NUMFUNC] NEQ 0
  629    0807   2              THEN
  630    0808   3                  BEGIN
  631    0809   3                  BIND
  632    0810   3                      CODMUL_TYP = .NICE + 2 : BBLOCK,
  633    0811   3                      FUNCTIONS = SID [SID$W_FUNCTIONS] : BITVECTOR [16];
  634    0812   3
  635    0813   3                  (.NICE) <0, 16> = NMA$C_PCCN_FCT;                      ! Place in Function paramter ID
  636    0814   3                  CODMUL_TYP [NMA$V_PTY_COD] = TRUE;                     ! Report is returned as a coded
  637    0815   3                  CODMUL_TYP [NMA$V_PTY_MUL] = TRUE;                     !   multiple.
  638    0816   3
  639    0817   3                  CODMUL_TYP [NMA$V_PTY_CLE] = .SID [SID$B_NUMFUNC];      ! 16 fields are permitted in the coded multi
  640    0818   3                  NICE = .NICE + 3;
  641    0819   3
  642    0820   3                  INCR INDEX FROM 0 TO SID$C_MAXFUNC - 1 DO
  643    0821   4                      BEGIN
  644    0822   4                      BIND
  645    0823   4                          FUN_TYP = .NICE : BBLOCK;
  646    0824   4
  647    0825   4                      IF .FUNCTIONS [.INDEX]
  648    0826   4                      THEN
  649    0827   5                          BEGIN
  650    0828   5                          FUN_TYP [NMA$V_PTY_COD] = TRUE;               ! Functions are returned as a coded value
  651    0829   5                          FUN_TYP [NMA$V_PTY_CLE] = 1;                  !   contained in 1 byte.
  652    0830   5                          (.NICE) <8, 8> = .INDEX;                      ! Place Function value in NICE message
  653    0831   5                          NICE = .NICE + 2;                            ! Advance to end of NICE buffer
  654    0832   4                          END;
  655    0833   3                      END;
  656    0834   2                  END;
```

```
 658   0835  2              !      Place in Hardware Address parameter
 659   0836  2              !      as a Hex Image 6
 660   0837  2              !
 661   0838  2              !          2 bytes      Hardware Address parameter ID
 662   0839  2              !          1 byte       Hardware Address type = Hex Image (HI-6)
 663   0840  2              !          1 byte       Image length = 6
 664   0841  2              !          6 bytes      Hardware Address value
 665   0842  2              !
 666   0843  2              !
 667   0844  2              (.NICE) <0, 16> = NMA$C_PCCN_HWA;
 668   0845  3                  BEGIN
 669   0846  3                  BIND
 670   0847  3                      TYPE = .NICE + 2 : BBLOCK;
 671   0848  3                  TYPE [NMA$V_PTY_NTY] = NMA$C_NTY_H;          ! returned as a Hex
 672   0849  3                  TYPE [NMA$V_PTY_NLE] = NMA$C_NLE_IMAGE;      !   image
 673   0850  3                  END;
 674   0851  2              (.NICE) <24,8> = SID$C_ADRLEN;                   !   of length 6
 675   0852  2              CH$MOVE (SID$C_ADRLEN, SID [SID$T_HRDWADR], (.NICE + 4) );
 676   0853  2              NICE = .NICE + 4 + SID$C_ADRLEN;                 ! Set pointer to end of buffer where next parameter
 677   0854  2
 678   0855  2
 679   0856  2              !
 680   0857  2              !      Place in Device Type parameter
 681   0858  2              !      as a coded value
 682   0859  2              !
 683   0860  2              !          2 bytes      Device Type parameter ID
 684   0861  2              !          1 byte       Device Type type = coded byte
 685   0862  2              !          1 byte       Device Type code
 686   0863  2              !
 687   0864  2              (.NICE) <0, 16> = NMA$C_PCCN_DTY;
 688   0865  3                  BEGIN
 689   0866  3                  BIND
 690   0867  3                      TYPE = .NICE + 2 : BBLOCK;
 691   0868  3                  TYPE [NMA$V_PTY_COD] = TRUE;                 ! Device Type is returned as a coded value
 692   0869  3                  TYPE [NMA$V_PTY_CLE] = 1;                    ! The coded value is 1 byte in length
 693   0870  3                  END;
 694   0871  2              (.NICE) <24, 8> = .SID [SID$B_DEVICE];
 695   0872  2              NICE = .NICE + 4;                                ! Set pointer to end of buffer where Elapsed Time will be pl
 696   0873  2
 697   0874  2              NICEBUF [DSC$W_LENGTH] = .NICE - .NICEBUF [DSC$A_POINTER];
 698   0875  2
 699   0876  2              RETURN TRUE;
 700   0877  1              END;                              ! Routine show_system


                                                      .PSECT  $PLIT$,NOWRT,NOEXE,2

                           45 43 41 52 54  00088 P.AAQ:  .ASCII  \TRACE\
                                           0008D          .BLKB   3
                               00000005    00090 P.AAP:  .LONG   5
                               00000000'   00094          .ADDRESS P.AAQ
        6D 65 74 73 79 73 5F 77 6F 68 73  00098 P.AAS:  .ASCII  \show_system\
                                           000A3          .BLKB   1
                               0000000B    000A4 P.AAR:  .LONG   11
                               00000000'   000A8          .ADDRESS P.AAS
```

```
                                                                        .PSECT   $CODE$,NOWRT,2

                                               01FC 00000 SHOW_SYSTEM:
                                                                        .WORD    Save R2,R3,R4,R5,R6,R7,R8
                              5E              10 C2 00002               SUBL2    #16, SP                           0655
                          0000' CF 9F 00005                             PUSHAB   P.AAR                             0678
                          0000' CF 9F 00009                             PUSHAB   P.AAP                             0677
                     0000G CF   01 DD 0000D                             PUSHL    #1
                          57     03 FB 0000F                            CALLS    #3, CNF$TRACE
                          56  04 AC 7D 00014                            MOVQ     SID, R7                           0698
                          56     68 3C 00018                            MOVZWL   (R8), NICE                        0680
                          66  04 A8 CO 0001B                            ADDL2    4(R8), NICE
                              78 8F 9B 0001F                            MOVZBW   #120, (NICE)                      0690
    02  A6         02     04     02 F0 00023                            INSV     #2, #4, #2, 2(NICE)               0694
                          A6     0F 8A 00029                            BICB2    #15, 2(NICE)                      0695
                          03  A6 06 90 0002D                            MOVB     #6, 3(NICE)                       0697
              04  A6      10  A7 06 28 00031                            MOVC3    #6, 16(R7), 4(NICE)               0698
                          56     0A CO 00037                            ADDL2    #10, NICE                         0699
                          66  82 BF 9B 0003A                            MOVZBW   #130, (NICE)                      0717
                          02  A6 CO 8F 88 0003E                         BISB2    #192, 2(NICE)                     0728
    02  A6         06     00     05 F0 00043                            INSV     #5, #0, #6, 2(NICE)               0729
                          16  A7 9F 00049                               PUSHAB   22(R7)                            0731
                          04  AE 9F 0004C                               PUSHAB   TIMBUF
              00000000G   00     02 FB 0004F                            CALLS    #2, SYS$NUMTIM
                          01     50 E8 00056                            BLBS     STATUS, 1$
                              04 AE 00059                               RET
                          03  A6 30 8A 0005A 1$:                        BICB2    #48, 3(NICE)                      0733
    03  A6         04     00     01 F0 0005E                            INSV     #1, #0, #4, 3(NICE)               0734
                          04  A6 04 AE 90 00064                         MOVB     TIMBUF+4, 4(NICE)                 0735
                          05  A6 80 8F 88 00069                         BISB2    #128, 5(NICE)                     0737
    05  A6         06     00     01 F0 0006E                            INSV     #1, #0, #6, 5(NICE)               0738
                          06  A6 02 AE 90 00074                         MOVB     TIMBUF+2, 6(NICE)                 0739
                          07  A6 30 8A 00079                            BICB2    #48, 7(NICE)                      0741
    07  A6         04     00     01 F0 0007D                            INSV     #1, #0, #4, 7(NICE)               0742
                          08  A6 06 AE 90 00083                         MOVB     TIMBUF+6, 8(NICE)                 0743
                          09  A6 30 8A 00088                            BICB2    #48, 9(NICE)                      0745
    09  A6         04     00     01 F0 0008C                            INSV     #1, #0, #4, 9(NICE)               0746
                          0A  A6 08 AE 90 00092                         MOVB     TIMBUF+8, 10(NICE)                0747
                          0B  A6 30 8A 00097                            BICB2    #48, 11(NICE)                     0749
    0B  A6         04     00     01 F0 0009B                            INSV     #1, #0, #4, 11(NICE)              0750
                          0C  A6 0A AE 90 000A1                         MOVB     TIMBUF+10, 12(NICE)               0751
                          56     0D CO 000A6                            ADDL2    #13, NICE                         0753
                          86  4E21 8F B0 000A9                          MOVW     #20001, (NICE)+                   0768
                          66  CO 8F 88 000AE                            BISB2    #192, (NICE)                      0777
          86      06      00     03 F0 000B2                            INSV     #3, #0, #6, (NICE)+               0778
                          66     30 8A 000B7                            BICB2    #48, (NICE)                       0780
          86      04      00     01 F0 000BA                            INSV     #1, #0, #4, (NICE)+               0781
                          86  1E A7 90 000BF                            MOVB     30(R7), (NICE)+                   0782
                          66     30 8A 000C3                            BICB2    #48, (NICE)                       0784
          86      04      00     01 F0 000C6                            INSV     #1, #0, #4, (NICE)+               0785
                          86  1F A7 90 000CB                            MOVB     31(R7), (NICE)+                   0786
                          66     30 8A 000CF                            BICB2    #48, (NICE)                       0788
          86      04      00     01 F0 000D2                            INSV     #1, #0, #4, (NICE)+               0789
                          86  20 A7 90 000D7                            MOVB     32(R7), (NICE)+                   0790
                          21  A7 95 000DB                               TSTB     33(R7)                            0806
```

```
                                    26  13 000DE          BEQL    4$
                            86  4E22 8F  B0 000E0          MOVW    #20002, (NICE)+                     0813
            86              06      66  C0 8F  88 000E5    BISB2   #192, (NICE)                        0815
                            00  21  A7  F0 000E9          INSV    33(R7), #0, #6, (NICE)+             0817
                                    50  D4 000EF          CLRL    INDEX                               0820
                    0C  22  A7      50  E1 000F1 2$:       BBC     INDEX, 34(R7), 3$                   0825
                            66  80  8F  88 000F6          BISB2   #128, (NICE)                        0828
            86              06      00  01  F0 000FA       INSV    #1, #0, #6, (NICE)+                 0829
                            86      50  90 000FF          MOVB    INDEX, (NICE)+                      0830
                            EB      0F  F3 00102 3$:       AOBLEQ  #15, INDEX, 2$                      0820
                            66  4E27 8F  B0 00106 4$:      MOVW    #20007, (NICE)                      0844
    02  A6              02  04      02  F0 0010B          INSV    #2, #4, #2, 2(NICE)                 0848
                    02  A6          0F  8A 00111          BICB2   #15, 2(NICE)                        0849
                    03  A6          06  90 00115          MOVB    #6, 3(NICE)                         0851
            04  A6      0A  A7      06  28 00119          MOVC3   #6, 10(R7), 4(NICE)                0852
                            56      0A  C0 0011F          ADDL2   #10, NICE                           0853
                            86  4E84 8F  B0 00122          MOVW    #20100, (NICE)+                     0864
                            66  80  8F  88 00127          BISB2   #128, (NICE)                        0868
            86              06      00  01  F0 0012B       INSV    #1, #0, #6, (NICE)+                 0869
                            86  24  A7  90 00130          MOVB    36(R7), (NICE)+                     0871
                            68      56  04  A8  A3 00134   SUBW3   4(R8), NICE, (R8)                  0874
                            50      01  D0 00139          MOVL    #1, R0                              0876
                                    04 0013C             RET                                         0877
```

; Routine Size: 317 bytes,   Routine Base: $CODE$ + 0250

```
: 702        0878  1 END                              ! End of module CNFSHOW
: 703        0879  0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
$PLIT$	172	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
OWN	16	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
$CODE$	909	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

| File | -------- Symbols -------- | | | Pages Mapped | Processing Time |
	Total	Loaded	Percent		
_255DUA28:[SYSLIB]STARLET.L32;1	9776	8	0	581	00:01.0
_255DUA28:[SHRLIB]NMALIBRY.L32;1	887	23	2	47	00:00.7

COMMAND QUALIFIERS

```
:     BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CNFSHOW/OBJ=OBJ$:CNFSHOW MSRC$:CNFSHOW/UPDATE=(ENH$:CNFSHOW)

: Size:           909 code + 188 data bytes
: Run Time:       00:23.1
: Elapsed Time:   00:39.7
: Lines/CPU Min:    2282
: Lexemes/CPU-Min: 19848
: Memory Used:  182 pages
: Compilation Complete
```

CNFSEND
LIS

CNFSTORE
LIS

NMLSHR
MAP

NPADEF
MDL

CNFSHOW
LIS

NMAHEAD
B32

NPAMAC
MAR

CNFWQDEF
LIS

NMATAIL
B32

NML

NMLDDL
B32

NML
MAP

NMLDEF
MDL

NMAFILES
LIS

NMAFIELDS
LIS

CNFWORKQ
LIS